

REMARKS

Applicant thanks the Examiner for pointing out and suggesting corrections for stylistic defects in the claims and for suggesting how the §112 problems can be addressed. These matters have all been addressed in the amended claims presented above. Accordingly, reconsideration of the rejections under 35 U.S.C. §112 is respectfully requested.

The Examiner has rejected claims 1-3 and 7-10 as unpatentable under 35 USC §103(a) in view of U.S. Patent No. 6,823,511 filed on January 10, 2000 by the inventors Paul E. McKenney and Brent Kingsbury (hereinafter “McKenney *et al.* patent”) in combination with U.S. Patent No. 6,353,869 filed on May 14, 1999 by the inventors Adi Ofer, Tuvia Leneman, and Natan Vishiltzky (hereinafter “Ofer, *et al.* patent”). The Examiner has rejected claims 4-6 and 11-13 as unpatentable under 35 U.S.C. §103(a) in view of the same McKenney *et al.* patent in combination with the same Ofer *et al.* patent and also U.S. Patent Application Publication No. 2001/0014905 filed by Tamiya Onodera on December 15, 2000 (hereinafter “Onodera application”). The Examiner has rejected claims 14 to 19 under 35 U.S.C. §103(a) in view of the same Ofer *et al.* patent in combination with U.S. Patent No. 6,748,593 filed on February 17, 2000 by Larry Bert Brenner and Luke Matthew Browning (hereinafter “Brenner *et al.* patent”).

Reconsideration of these rejections, and allowance of claims 1-19 as amended, is respectfully requested for the following reasons.

A. Claims 1-13

First, with respect to the rejection of the claims 1-13, the teachings of the McKenney *et al.* patent and the teachings of the present application, as defined in these claims, do not correspond. With reference in particular to Figure 6 of the present application (see the replacement Figure 6 attached to this document), the present invention (in a “mutex” embodiment) constructs a single synchronization object for a multi-computer system through the combined use of a single global mutex 43 operating in a global memory segment 42 in conjunction with a plurality of local mutexes 66 each operating in the local memory 38 of an individual node 10 (a “node” is a separate multiprocessor platform). Thus, the single synchronization object is managed by the one global mutex 43 and by as many local mutexes 66 as there are multiprocessor platforms or nodes 10, 20, etc. (Figure 1) having threads that

are competing for the synchronization object. In addition, the present invention specifies that each of the local mutexes 66 is managed by the corresponding node's local operating system, whereas the global mutex is managed by separate software that is not built into a conventional operating system (such as Microsoft's Windows NT, Version 4.0, used in the disclosed embodiments). The present invention thus contemplates a cooperative arrangement between a single global mutex and a plurality of local mutexes managed by the local operating systems, each contributing in a complimentary to a highly advantageous and balanced arrangement, as is fully described in the application.

McKenney *et al.*, contrary to this, teaches constructing a synchronization object for a multi-computer system without providing any single, global lock device. Instead, McKenney *et al.* teaches achieving a synchronization object by providing each node (or multiprocessor platform) with several local lock devices, one lock device for each group of several local processors. There is no one truly global lock device working in cooperation with local lock devices, contrary to the teachings of the present application.

More specifically, and with reference to the McKenney *et al.* patent, McKenney *et al.* discloses an array of four multiprocessor platforms or "nodes" (referred to as "node 1", "node 2", "node 3", and "node 4") each of which "nodes" contains four individual processors. Thus, there are 16 processors total. Within each node, McKenney *et al.* organize the processors into two "groups" of two processors each. Thus, node 1 contains processor group 0 and processor group 1; node 2 contains processor group 2 and processor group 3; node 3 contains processor group 4 and processor group 5; and node 4 contains processor group 6 and processor group 7. McKenney *et al.* then disclose the implementation of 16 individual locks. These locks are named "lock a", "lock b", "lock c", ... , and "lock p". With reference to Figure 5 of the McKenney *et al.* patent, it can be seen that the "lock a", for example, is implemented by means of eight "global structures 81", one for each of the processor groups 0 through 7. These eight global structures are identified individually in Figure 5 by the lock identification letter "a" written under a group identification number "0", "1", ... , or "7" as follows: "a" under "0" and "a" under "1" in the "node 1 page" of global memory; "a" under "2" and "a" under "3" in the "node 2 page" of global memory; "a" under "4" and "a" under "5" in the "node 3 page" of global memory; and "a" under "6" and "a" under "7" in the "node 4 page" of global memory). Each of these "global structures" includes or points to the structures illustrated in Figure 6 and described in the Table 1 (column 7), all of which is

explained in the description presented in col. 7, line 29 through col. 8, line 26 of the McKenney *et al.* patent.

Thus, eight “global structures 81” comprise the exemplary “lock a”. All of these structures are local to a group of two processors, and none is “global” with respect to the others. Accordingly, there is no single, global, control device cooperating with lesser, operating-system-managed control devices to be found in the McKenney *et al.* patent.

Independent claim 1, in contrast to this, calls specifically for “global memory” having a “spinlock” and “a data structure” “wherein one or more records for global synchronization objects may be established.” The “synchronization software system” then uses “the above spinlock and the data structure” to “resolve requests for the synchronization object as between threads residing on different nodes.” The “system” also uses “local synchronization objects created by the local operating systems on nodes having threads awaiting access to resolve requests for the synchronization object as between threads residing on the same node.” Thus, independent claim 1, and also independent claim 8 which is similarly worded, both call for the use of a global synchronization object to resolve inter-node contention between threads running on different nodes, and they call for the use of multiple, operating-system managed, local synchronization objects to then resolve intra-node contention between threads running on one or more processors within a single node or multiprocessor platform. Clearly, this arrangement is entirely different than that described in the McKenney *et al.* patent, where there is no truly global synchronization object, and where several synchronization objects of equal status are present on each of several multiprocessor platforms.

Accordingly, McKenney *et al.* by itself does not teach the structure called for by these claims and described in the present application. The Examiner says that the Ofer *et al.* patent adds to McKenney *et al.* patent’s teachings “data structure including provision for recording in a queue the identity of nodes having threads awaiting access to the synchronization object/and queue of node identities to resolve requests for the synchronization object as between threads residing on different nodes” (Office Letter mailed 01/26/2005, page 5, paragraph 9) With reference to Figure 1 of Ofer *et al.*, it discloses the “n” processors “1a” through “1n” connected by what the specification describes as SCSI (small computer system interface) busses to shared memory 2 and a shared device 4 (for example, a disk storage

device). Only a single “lock request queue 46” is shown (in the shared memory 2), not a global control device that is supported by multiple local control devices as is taught by the present application. Likewise, the Onodera published application (which the Examiner has cited against the claims 4-6 and 11-13 in addition to the other two references) describes what appears in Onodera’s Figure 1 to be only one processor with only one mutex control arrangement, and not a global arrangement and separate local arrangements all working together as described and claimed in the present application. Clearly, these three references, taken either alone or together in various combinations, do not teach the present invention as it is defined in independent claims 1 and 8.

Accordingly, independent claims 1 and 8 are in condition for allowance. And dependent claims 2-7 and 9-13 are also in condition for allowance, since they are dependent upon allowed claims. Allowance of claims 1-13 is therefore respectfully requested.

B. Claims 14-19

The two claim sets 14-16 and 17-19 differ from each other only in that the claim set 14-16 contains method claims while the claim set 17-19 contains “software computer program” claims. Thus, independent claims 14 and 17 contain essentially the same limitations, and this discussion can focus upon independent claim 14 as representative of the similar independent claim 17.

Independent claim 14 is actually quite similar to independent claim 1 in what it requires. It requires a global synchronization object having a spinlock mechanism. A thread may gain control of this object if it is free. But if a thread does not gain control of this object, then this claim has the thread “seek ownership of a local synchronization object established on the thread’s node by a local operating system,” and temporarily any “threads on the thread’s node” are blocked “from seeking ownership of the local synchronization object” and are forced “into suspension.” This claim goes on, in its last paragraph, to describe what happens when a thread releases the global synchronization object, but there is no need to repeat that here. But it should already be clear to the Examiner that independent claim 14, and the similar independent claim 17, both require the use of a global synchronization object and also the use of multiple local synchronization objects, one on every node (or

multiprocessor platform) where threads seeking the global object are active, the local objects managed by the local operating systems, as was the case with claims 1-13.

None of the prior art patents reviewed above include this combination of features, as has been explained in the discussion of independent claim 1 presented above. The Examiner has cited an additional patent against claims 14-19, the Brenner *et al.* patent. So it remains to be determined whether the Brenner *et al.* patent adds any teachings that, either alone or in combination with the teachings of the Ofer *et al.* patent, anticipates this claim language.

The Brenner, *et al.* patent discloses, in Figure 1, a multiprocessor system including three nodes each having two “CPUs” or processors. It teaches “load balancing,” not “synchronization” and “access control.” It does this by moving threads from one processor to another to balance the processor loads and to thereby insure that some threads are not “starved,” or rarely ever given access to processor time. The patent teaches establishing separate “global” and “local” queues of suspended threads and then attempting to optimize the execution of these threads by assigning their execution to the least busy processors. But the focus of this patent’s teachings is not upon synchronization objects and their implementation. Thus, this patent appears to be generally irrelevant to the present invention.

There is one point in the Brenner, *et al.* patent where it is explained that before attempting to access a queue of suspended threads for the purpose of dispatching a thread, “the dispatcher 150 attempts to lock the run queue.” If the “global run queue” of suspended threads is “locked” because “another CPU has locked” it, the patent teaches that the dispatcher should “not retry the attempt to lock the global run queue, but instead” should select “the local run queue” and attempt “to dispatch a thread from it.” (Brenner *et al.* patent, col. 4, lines 1-12) This is the passage which the Examiner has cited. But this passage says nothing about the nature of these locks – whether they are implemented by a global synchronization object used in conjunction with local synchronization objects managed by the local operating systems, as all of the claims before the Examiner require.

Accordingly, the Brenner *et al.* patent by itself teaches nothing about how to design a global synchronization object – the Brenner *et al.* patent simply uses such a synchronization object and does not describe its implementation. Likewise, the Ofer patent, as was explained more fully above, introduces only one global synchronization object. It does not teach

providing separate global and local objects working together in a synergistic manner, with the local objects being managed by the local operating systems. Independent Claims 14 and 17 are accordingly allowable over the art of record. The dependent claims 15-16 and 18-19 are also in condition for allowance, since they are dependent upon claims which are in condition for allowance.

Accordingly, all of the claims 1-19 presently before the Examiner are in condition for allowance. Early and favorable action is, accordingly, respectfully requested.

C. Conclusion


Applicants believe that the present application, as amended, is now in condition for allowance. Early and favorable reconsideration and allowance of this application, as amended, is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§1.16-1.17, or credit any overpayment, to Deposit Account No. 08-2025. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 08-2025. If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicants hereby petition for such extension under 37 C.F.R. §1.136 and authorizes payment of any such extensions fees to Deposit Account No. 08-2025.

Respectfully submitted,

Date 4/26/2005
FOLEY & LARDNER LLP
321 North Clark Street, Suite 2800
Chicago, Illinois 60610

By 
James A. Sprowl
Attorney for Applicants
Registration No. 25,061

Telephone: (312) 832-4586
Facsimile: (312) 832-4700

Telephone 847-446-7399

Amendments to the Drawings

Seven sheets of corrected replacement drawings, and one new formal drawing (Figure 14), are submitted herewith. These eight sheets of drawings are appended to the end of this document. Applicant respectfully requests that these eight sheets of drawings replace the corresponding drawings presently of record.

Figure 14 is a new formal drawing that corresponds to the drawing sketch labeled Figure 14 filed with the original application on its filing date. This formal drawing was not published as part of the published application, and it appears to have become lost from the USPTO's file wrapper. Its submission at this time does not constitute the submission of any new matter.

The seven replacement drawings all contain corrections (described below). One of the corrections made to Figure 4 conforms that drawing to the specification and corrects a typographic error, and some corrections add missing reference numbers to the drawings. The other corrections made to the remaining six replacement drawings simply conform the formal set of drawings to the informal drawing sketches filed with the original application on its filing date, and accordingly these corrections do not introduce any new matter.

Figure 1 – The general reference number “1” has been added to indicate the entire contents of this drawing. Two vertical lines have been added: one connecting the box 46 to the box 34, and one connecting the box 48 to the box 36. These corrections conform formal Figure 1 to the drawing sketch Figure 1. In addition, four additional vertical lines have been added: Two vertical lines connecting the two data busses 50 and 52 to the node 10's memory and I/O controller 34; and two vertical lines connecting the two data busses 50 and 52 to the node 20's memory and I/O controller 36. These four added vertical lines conform Figure 1 to the description presented in the paragraph [0047], where it says: “One or more data busses 50 and 52 interconnect the memory and I/O controllers 34 and 36 such that all the nodes 10 and 20 may access the same shared global memory segments 42 and 44.” Applicant respectfully requests the Examiner's approval of this change.

Figure 3 – In the box 60, “(FIGS. 7-17” has been changed to “(FIGS. 7-14)”. This correction conforms formal Figure 3 to the drawing sketch Figure 3.

Figure 4 – In the box 400, “GLOBAL MVTEX DATA RECORD” has been changed to “GLOBAL MUTEX DATA RECORD”. This correction conforms formal Figure 4 to the drawing sketch Figure 4. In the box 410, “GLOBAL WAST QUEUE” has been changed to “GLOBAL WAIT QUEUE”. This correction causes Figure 4 to agree with the text presented in paragraph 81, where the following phrase appears: “... The record 400 includes a global wait queue 410, which links”

Figure 5 – In the box 506, “LOCAL MVTEX” has been changed to “LOCAL MUTEX”. This correction conforms formal Figure 5 to the drawing sketch Figure 5.

Figure 6 – In the box 66, “LOCAL MVTEX” has been changed to “LOCAL MUTEX”. The reference numeral “1000” has been added to the box containing the legend: “RELEASE A MUTEX (FIG. 10)” to conform this figure to Figure 10. The reference numeral “400” has been added to the box containing the legend: “GLOBAL MUTEX DATA RECORD (FIG. 4)” to conform this figure to Figure 4. The reference numeral 410 has been added to the box containing the legend “GLOBAL WAIT QUEUE” to further conform this figure to Figure 4. The reference numeral “38” has been added to the largest dashed line box in this figure which is labeled: “NODE 10 LOCAL MEMORY”. Arrows have been added connecting the three boxes labeled 602, 604, and 606, the arrows matching those connecting the other three boxes 414, 416, and 418. The direction of the two arrows connecting the boxes 800 and 1200 to the box 56 now point away from the box 56. The four reference numbers 980, 910, 912, and 914 have been added to the box labeled “SUSPEND” to indicate that the “suspend” step is implemented by these four steps shown in Figure 9, as is fully explained in paragraph [00099] of the specification – this conforms Figure 6 to the text in paragraph [00099]. Unless otherwise indicated above, these corrections conform formal Figure 6 to the drawing sketch Figure 6.

Figure 7 – The general reference numeral “700” has been added to indicate the entire contents of this drawing. The reference numeral “508” has been corrected to read “708”. These corrections conform formal Figure 7 to the drawing sketch Figure 7.

Figure 9 – In the box 915, “... NODE FROM ...” has been changed to “... NODE, AND IF COUNT = 0 REMOVE THE NODE FROM ...”. In the box 916, “... NODE ID LOCATION 406 OF ...” has been changed to “... NODE ID IN THE OWNER NODE ID LOCATION 406 OF ...”. In the box 912, “HAVE THE LOCAL O.P. 62 CAUSE ...” has been changed to “HAVE THE LOCAL O.S. 62 CAUSE ...”. These corrections conform formal Figure 9 to the drawing sketch Figure 9.

Figure 14 – Applicant believes that a formal drawing corresponding to the sketch Figure 14 (filed with the original application) was delivered to the USPTO along with the other 13 sheets of formal drawings, but formal Figure 14 appears to have become lost from the file. Accordingly, applicant herewith submits a new formal Figure 14 that corresponds in all respects to the sketch Figure 14 filed with this application on its filing date. In addition, the question “ANOTHER NODE HAS NOT BEEN NOTIFIED?” which appeared in the box 1110 of the sketch Figure 14 has been rewritten to “ANOTHER NODE IS AVAILABLE THAT HAS NOT BEEN NOTIFIED?” to improve its clarity and to conform this language to the language which appears in paragraph [00108] of the specification. Since this revised language is copied directly from and is fully supported by the specification, this change introduces no new matter.